



The bridge to possible

How to become an IOS XE Terraform Expert

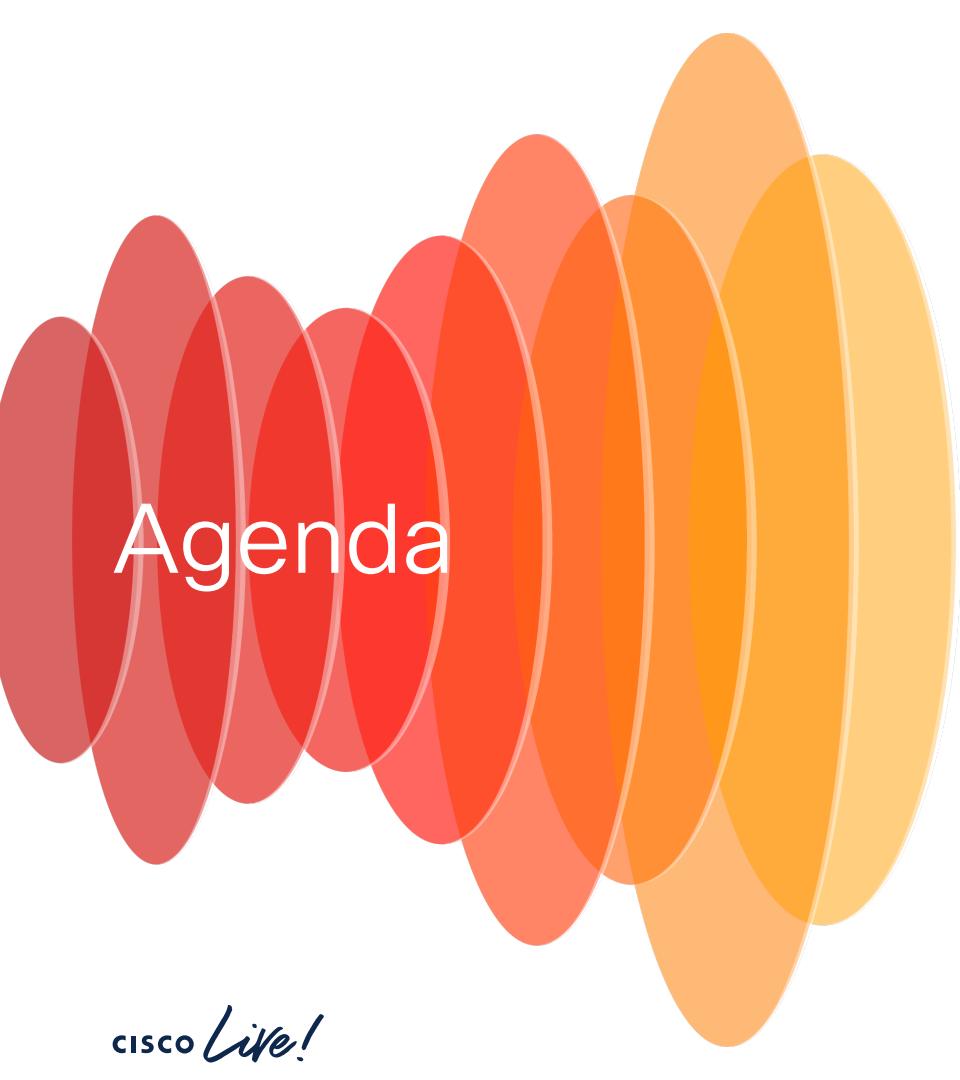
DEVWKS-2042

Story DeWeese, Technical Marketing
@StoryDeWeese
Cisco IOS XE & Terraform

cisco *Live!*

#CiscoLive

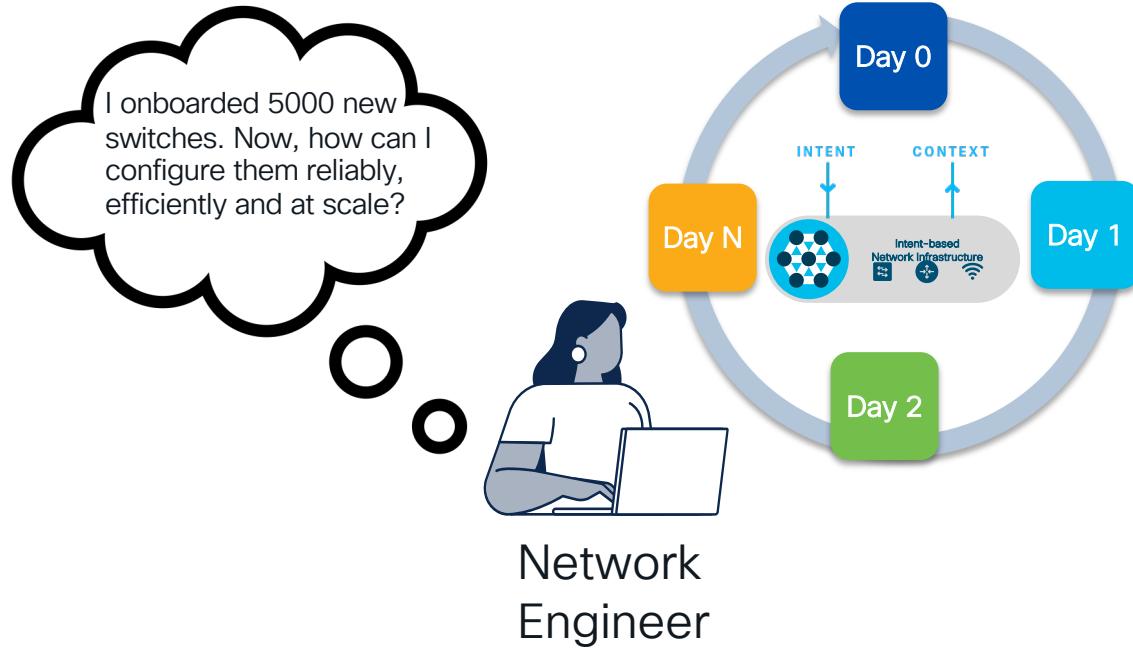
Infrastructure as Code (IAC) allows us to automate provisioning and management of Cisco Catalyst 9000 devices more efficiently. In this session, automation tooling will be used to programmatically for network config. Infrastructure as Code concepts will be demonstrated on the Catalyst 9000 using Terraform. Both enterprise campus and cloud-native use cases and examples will be shared. You'll leave this session with ideas to help you get started with Infrastructure as Code, understand benefits of IAC, and understand more about the Terraform tooling. With IaC and Cisco IOS XE, you can automate any tooling on any interface.



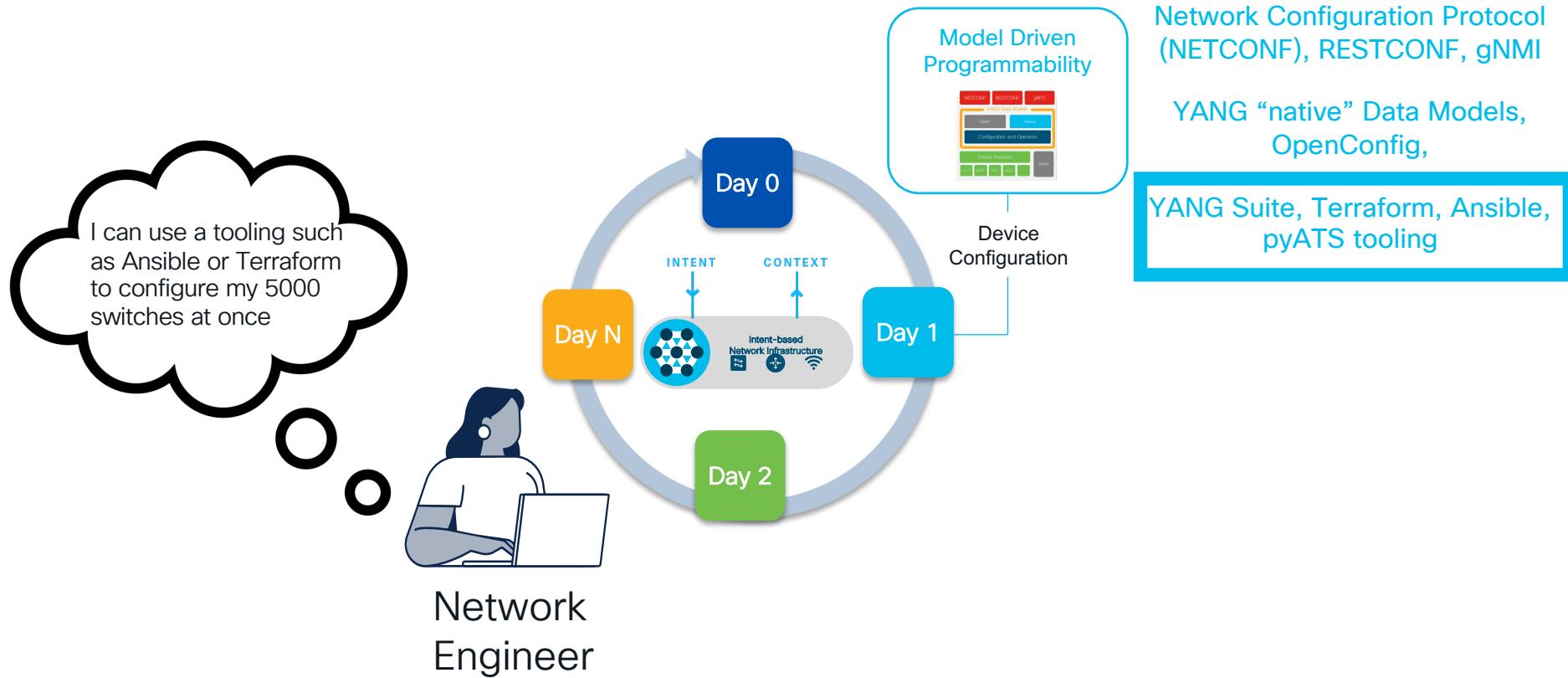
Agenda

- Programmability and Automation
- Terraform
 - Pre-req's
 - Install the Terraform Tooling
 - Turn on RESTCONF on switch + AAA + User + NACM
 - Service-level ACL
- Use Case Examples
- Hands On!
- Resources

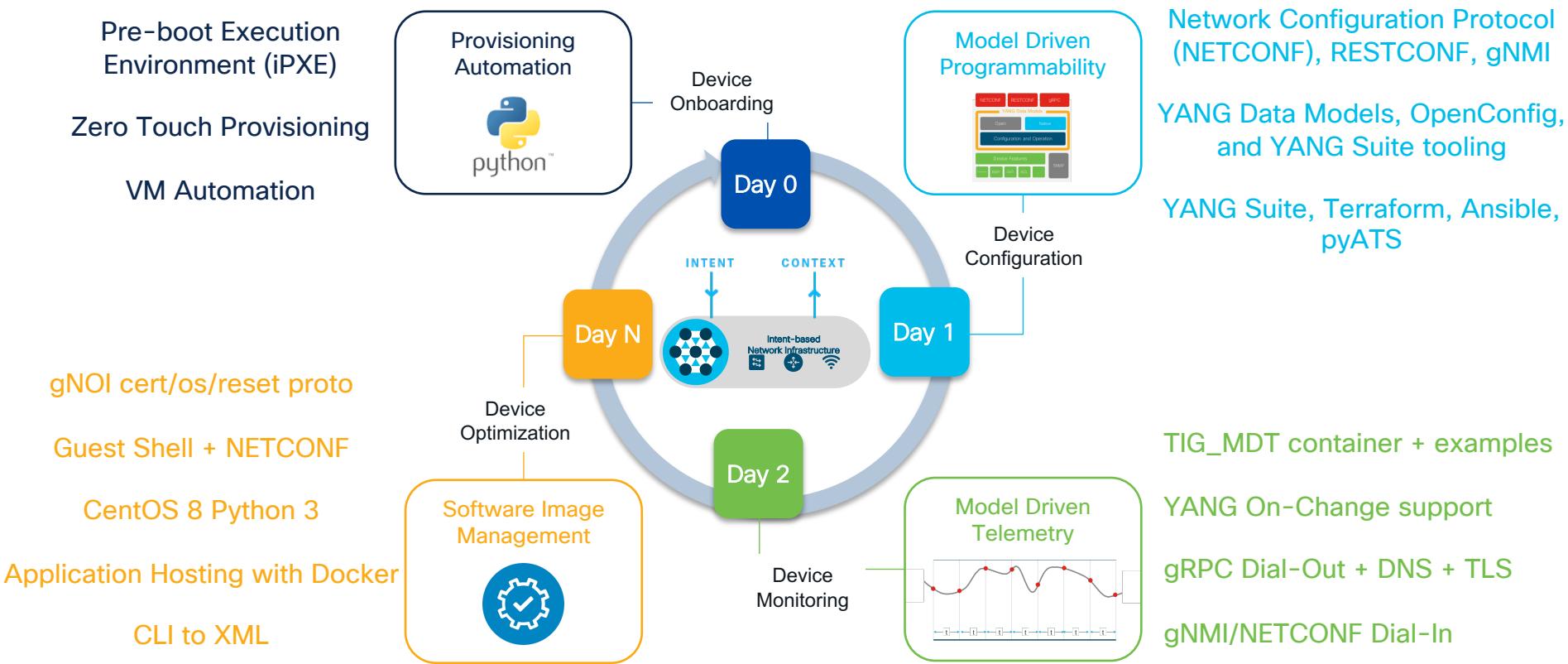
Cisco IOS XE Programmability & Automation Lifecycle



Cisco IOS XE Programmability & Automation Lifecycle



IOS XE Programmability & Automation Lifecycle



IOS XE Programmability and Telemetry “Stack”

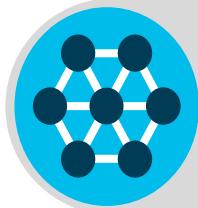
CLI

SNMP

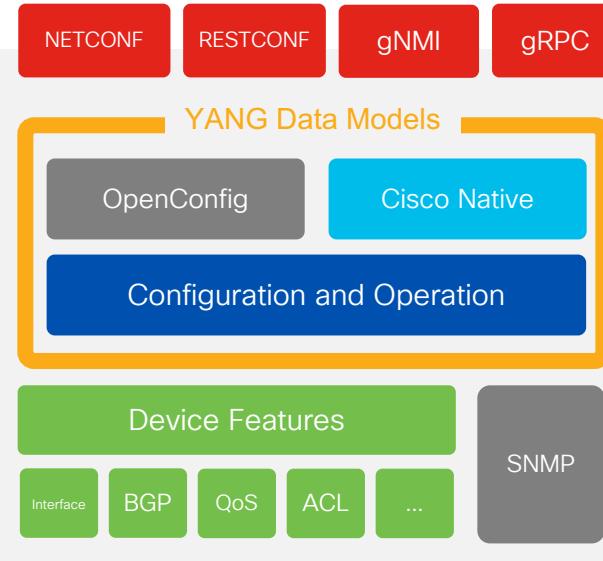
WebUI

The NETCONF, RESTCONF, gNMI and gRPC are programmatic interfaces that provide additional methods for interfacing with the IOS XE device – Just like the CLI, SNMP, and WebUI is used for configuration changes and operational metrics so can the programmatic interfaces of NETCONF, RESTCONF, gNMI, and gRPC.

YANG data models define the data that is available for configuration and streaming telemetry



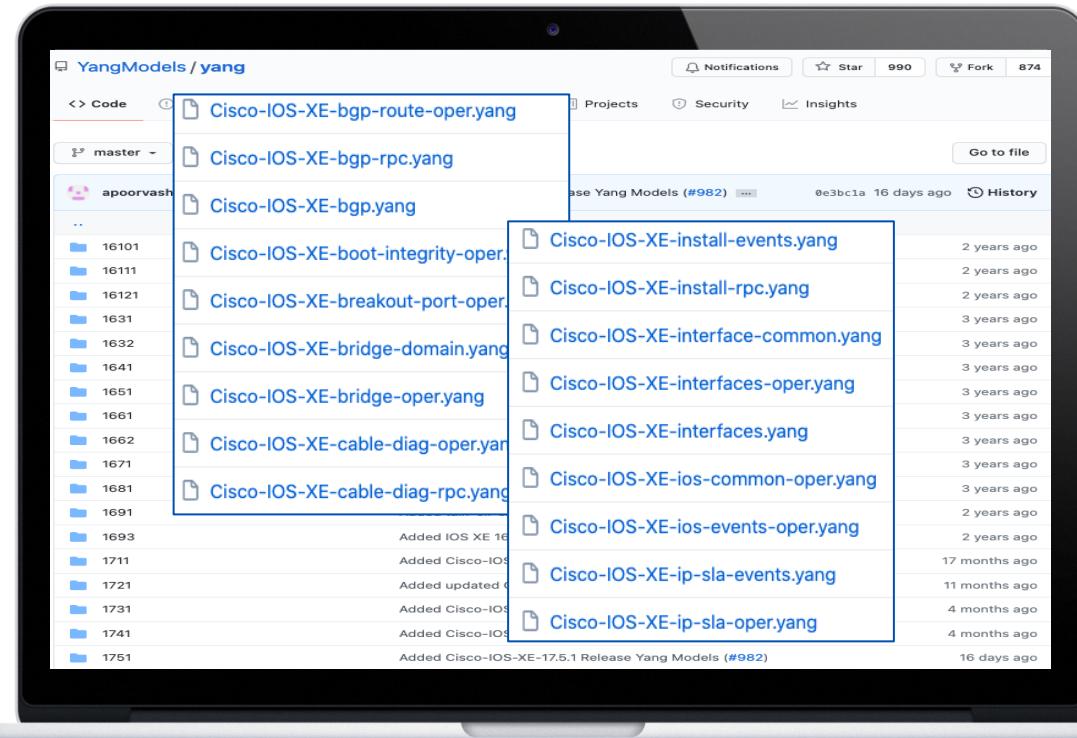
Intent-based Network Infrastructure



IOS XE - YANG Model Coverage on GitHub

RFC7950 states that “YANG is a data modeling language used to model configuration data, state data, Remote Procedure Calls, and notifications for network management protocols”

YANG module name.yang	Description
Cisco-IOS-XE-native	running-config
Cisco-IOS-XE-{feature}-cfg	Feature configuration
Cisco-IOS-XE-{feature}-oper	Feature operational data
Cisco-IOS-XE-{feature}-rpc	Actions
Cisco-evpn-service	EVPN service abstraction
OpenConfig-{feature}	abstraction for config & oper



<https://github.com/YangModels/yang/tree/master/vendor/cisco/xe>

RBAC/NACM for APIs – Enable read-only user

CLI to populate NACM with Read Only rules:

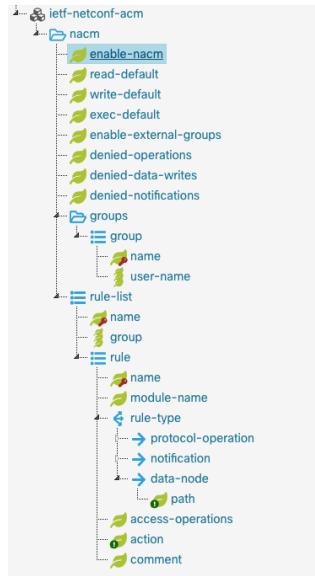
```
request platform software yang-management nacm populate-read-rules privilege 1
```

CLI to reset NACM:

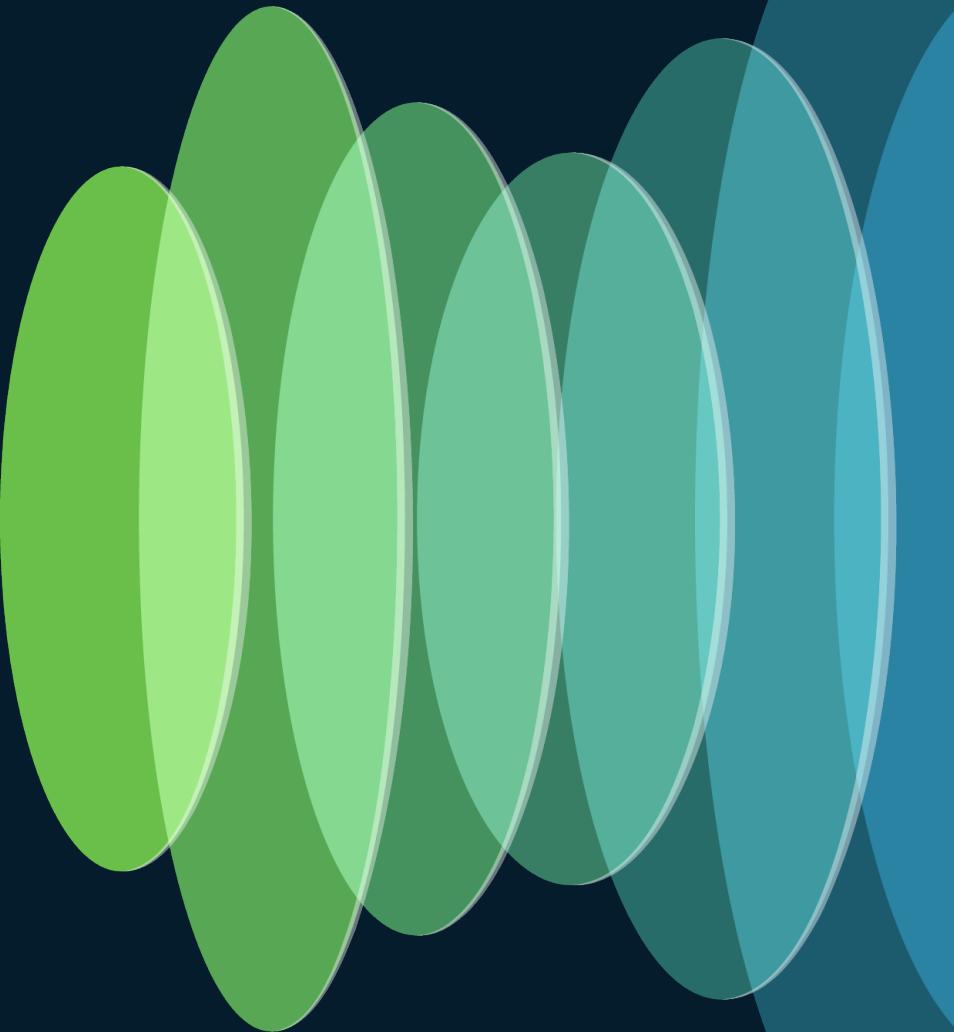
```
request platform software yang-management nacm reset-config
```

- The NACM rules are part of the “ietf-netconf-acm” YANG data model and are not part of the running-configuration
- NACM configuration is persistent across reloads as it is part of YANG DMI
- The NACM rules can be read by performing a GET or READ operation against the NACM xpath. They can also be reset by CLI
- The “populate read rules” CLI can be used to enable read-only operations for priv1

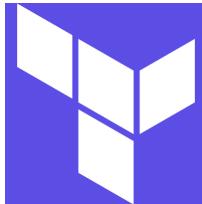
```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
<enable-nacm>true</enable-nacm>
<read-default>deny</read-default>
<write-default>deny</write-default>
<exec-default>deny</exec-default>
<enable-external-groups>true</enable-external-groups>
<groups>
<group>
<name>gnmi-read-only-users</name>
<user-name>gnmi_reader</user-name>
</group>
</groups>
<rule-list>
<name>admin</name>
<group>PRIV15</group>
<rule>
<name>permit-all</name>
<module-name></module-name>
<access-operations></access-operations>
<action>permit</action>
</rule>
</rule-list>
<rule-list>
<name>gnmi-read-only-rules</name>
<group>gnmi-read-only-users</group>
<rule>
<name>deny-write-ops</name>
<module-name></module-name>
<access-operations>create update delete exec</access-operations>
<action>deny</action>
</rule>
<rule>
<name>allow-read</name>
<module-name></module-name>
<access-operations>read</access-operations>
<action>permit</action>
</rule>
</rule-list>
</nacm>
```



Terraform



Terraform uses the RESTCONF API on IOS XE



Infrastructure as Code (IaC) Software Tool providing a consistent CLI workflow to manage hundreds of cloud services. Terraform codifies cloud APIs into declarative configuration files.

Resources:

Ask IOS XE Terraform Provider Webex space:

<https://eurl.io/#PtsT8eJFI>

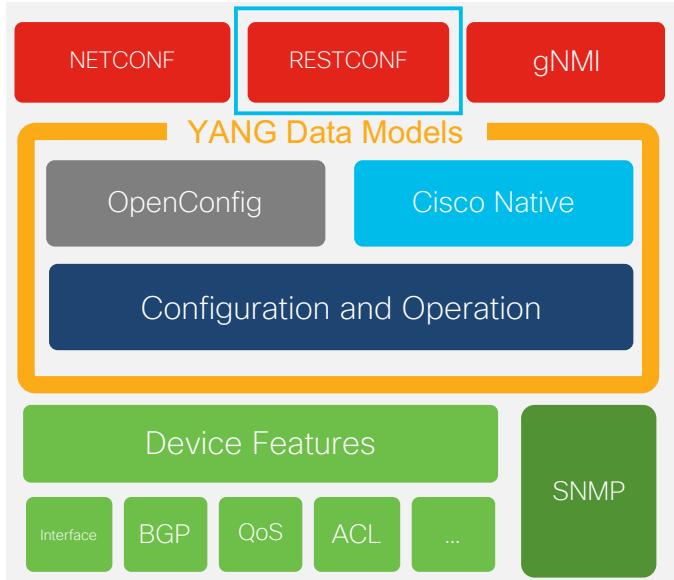
GitHub Repository:

<https://github.com/CiscoDevNet/terraform-provider-iosxe/>

Provider Documentation:

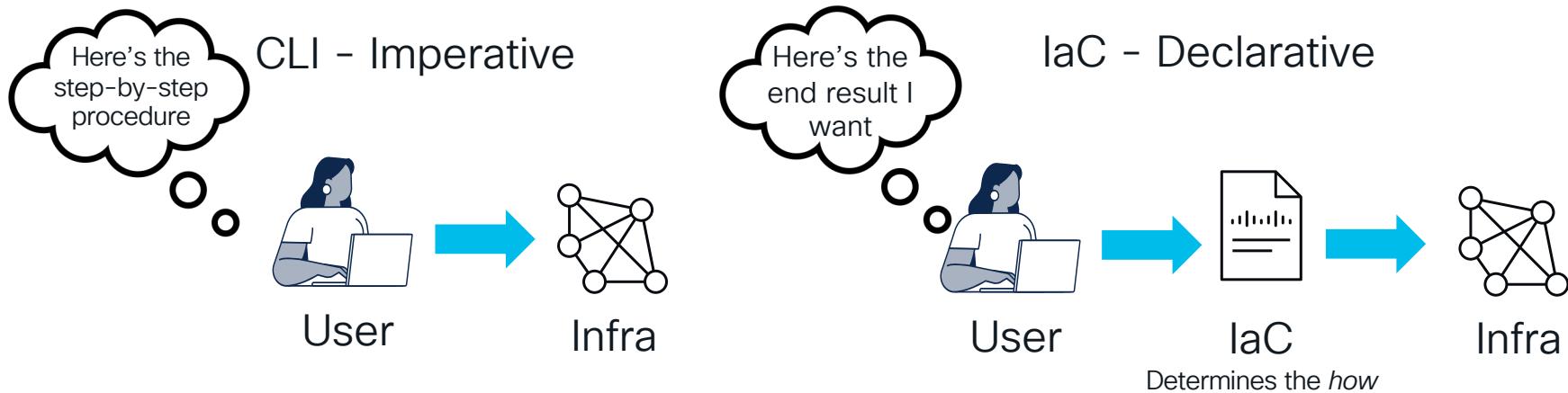
<https://registry.terraform.io/providers/CiscoDevNet/iosxe/latest>

Terraform uses the RESTCONF API



What's IaC?

The imperative CLI manages configuration using a step-by-step procedure, while the declarative Infrastructure as Code (IaC) manages configuration through the end result or state.



Learn more about IaC here:

<https://developer.cisco.com/iac/#:~:text=Adopting%20Infrastructure%20as%20Code%20allows.data%20center%20to%20the%20edge>

What's different about Terraform?

State-FUL

- Checks the current configuration before making changes to help avoid any potential conflicts
- Quick to remove config, no need to create a separate file

```
$ terraform destroy
```

Cisco IOS XE API Pre-Req

```
Cat9k-1#conf t
Enter configuration commands, one per line. End with
CNTL/Z.
Cat9k-1(config)#aaa new-model •                                     Enable AAA
Cat9k-1(config)#aaa authentication login default local
Cat9k-1(config)#aaa authorization exec default local
Cat9k-1(config)#username admin privilege 15 password cisco

Cat9k-1(config)#restconf •                                         Enable RESTCONF
```

This is the most basic example 😊
There are additional configuration knobs for ACL, AAA, RBAC, etc

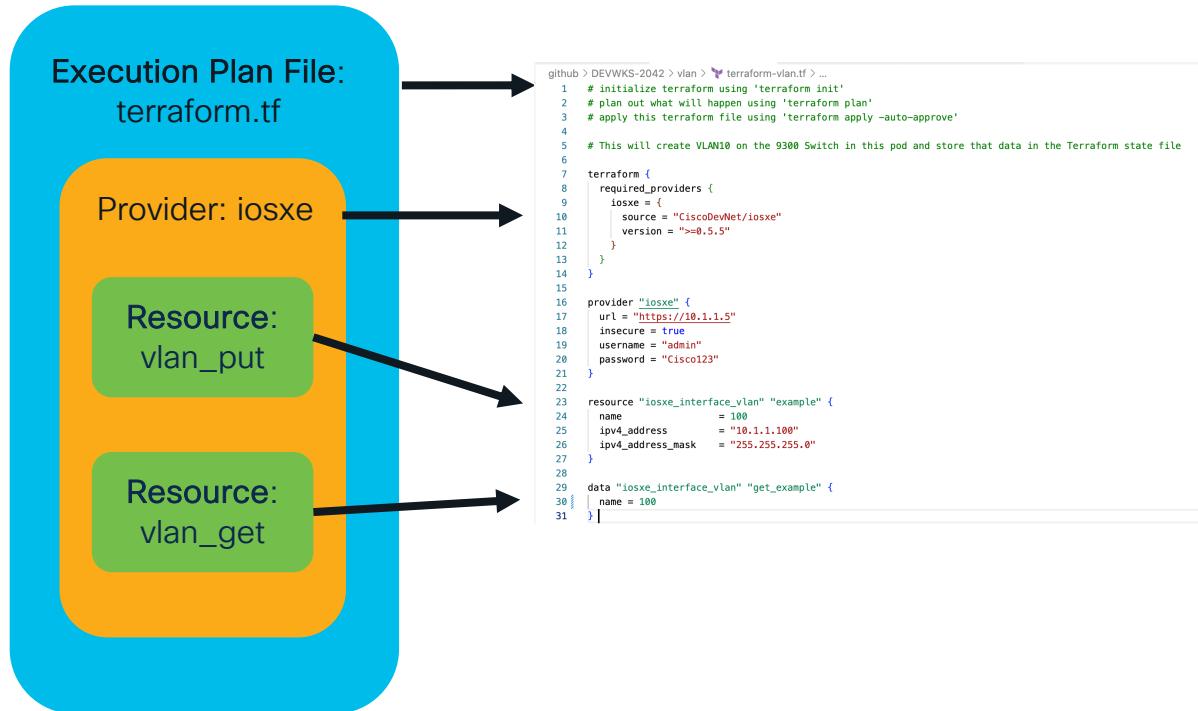
Terraform Terminology

Terraform uses an execution plan file with a provider and resource definitions.

An execution plan file defines the provider and resources. It is written in HashiCorp Configuration Language (HCL), similar to JSON, and stored with a .tf extension

A provider is a plugin to make a collection of resources accessible

A resource (or infrastructure resource) describes one or more infrastructure objects managed by Terraform. With the IOS XE Terraform provider, resources can be considered the same as a configurable feature



Getting Started with Terraform + IOS XE Provider

1. Enabling the RESTCONF API on the switch

```
Switch# conf t  
Switch(config)# restconf
```

2. Install [Terraform](#)

```
$ apt install terraform
```

3. Clone the [IOS XE Terraform Provider](#) GitHub repository

```
$ git clone https://github.com/CiscoDevNet/terraform-provider-iosxe  
... or ...  
$ https://github.com/jeremycohoe/cisco-ios-xe-dcloud-terraform
```

4. Apply Terraform VLAN example

```
$ terraform init  
$ terraform plan  
$ terraform apply terraform.tf
```

Terraform One Slide

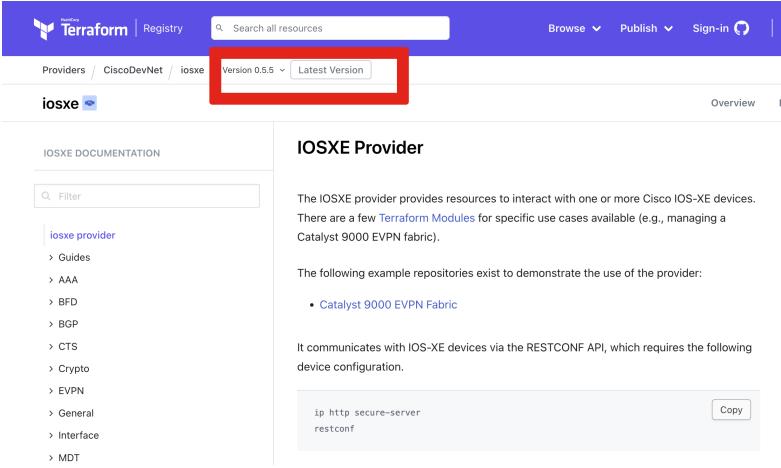
- **HCL** (HashiCorp Configuration Language) - Establishes the syntax Terraform uses for things like arguments, blocks, literal values, and expressions, and writing plans.
- **Provider** - Plugins responsible for understanding API interactions with other platforms and exposing resources based on their APIs.
- **Data Source** - Allows Terraform to use (read) information defined outside of Terraform. Example: providers, local-only
- **Resource** - Are the most important element in the Terraform language. Each resource block describes one or more infrastructure objects - devices, interfaces, operations.
- **Init** - The command is used to initialize a working directory containing Terraform configuration files. This is the first command that should be run, and it is safe to run this command multiple times. It will install the required providers and modules.
- **Plan** - compares the managed infrastructure state to the configuration, and it determines which changes are necessary. It presents a human-readable summary to the user.
- **Apply** - Makes changes to real infrastructure in order to make it match the desired state. It may use saved plans or creates a new plan and asks for approval.

Ref: <https://www.terraform.io/docs>

IOS XE Terraform Provider

Documentation and details about the provider are available on the Hashicorp Registry
<https://registry.terraform.io/providers/CiscoDevNet/iosxe/latest>

Source code is in the GitHub Repository: <https://github.com/CiscoDevNet/terraform-provider-iosxe/>



The screenshot shows the Hashicorp Registry interface. On the left, there's a sidebar with navigation links like 'Providers' (selected), 'CiscoDevNet', 'iosxe', 'iosxe DOCUMENTATION', and a 'Filter' dropdown. The main content area has a header 'IOSXE Provider' with a sub-header 'The IOSXE provider provides resources to interact with one or more Cisco IOS-XE devices.' Below this, there's a section for 'Example repositories' with a link to 'Catalyst 9000 EVPN Fabric'. At the bottom, there's a snippet of configuration code: 'ip http secure-server' and 'restconf'.

iosxe provider

- > Guides
- > AAA
- > BFD
- > BGP
- > CTS
- > Crypto
- > EVPN
- > General
- > Interface
- > MDT

MDT

MPLS

Management

Multicast

OSPF

QoS

Routing

Switching

System

VRF

General

Resources

- iosxe_cli**
- iosxe_restconf**
- iosxe_save_config**

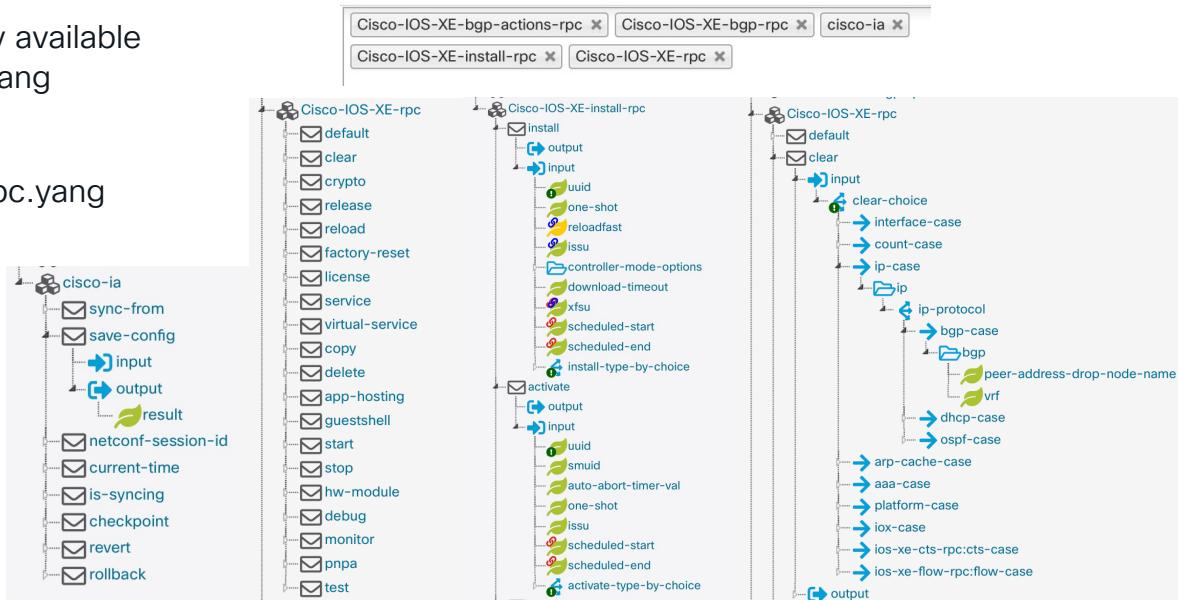
Data Sources

- iosxe_restconf**

RPC Actions - generic provider

Terraform IOS XE Provider Roadmap includes various RPCs

1. CLI RPC – already generally available
2. Cisco-IOS-XE-install-rpc.yang
3. Cisco-ia.yang
4. Cisco-ios-xe-rpc.yang
5. Cisco-ios-xe-wireless-*-rpc.yang



Track Updates here:

<https://github.com/CiscoDevNet/terraform-provider-iosxe/issues/126>

Install use case

Clear ip bgp example

Terraform resource utilizing the CLI RPC

<https://registry.terraform.io/providers/CiscoDevNet/iosxe/latest/docs/resources/cli>

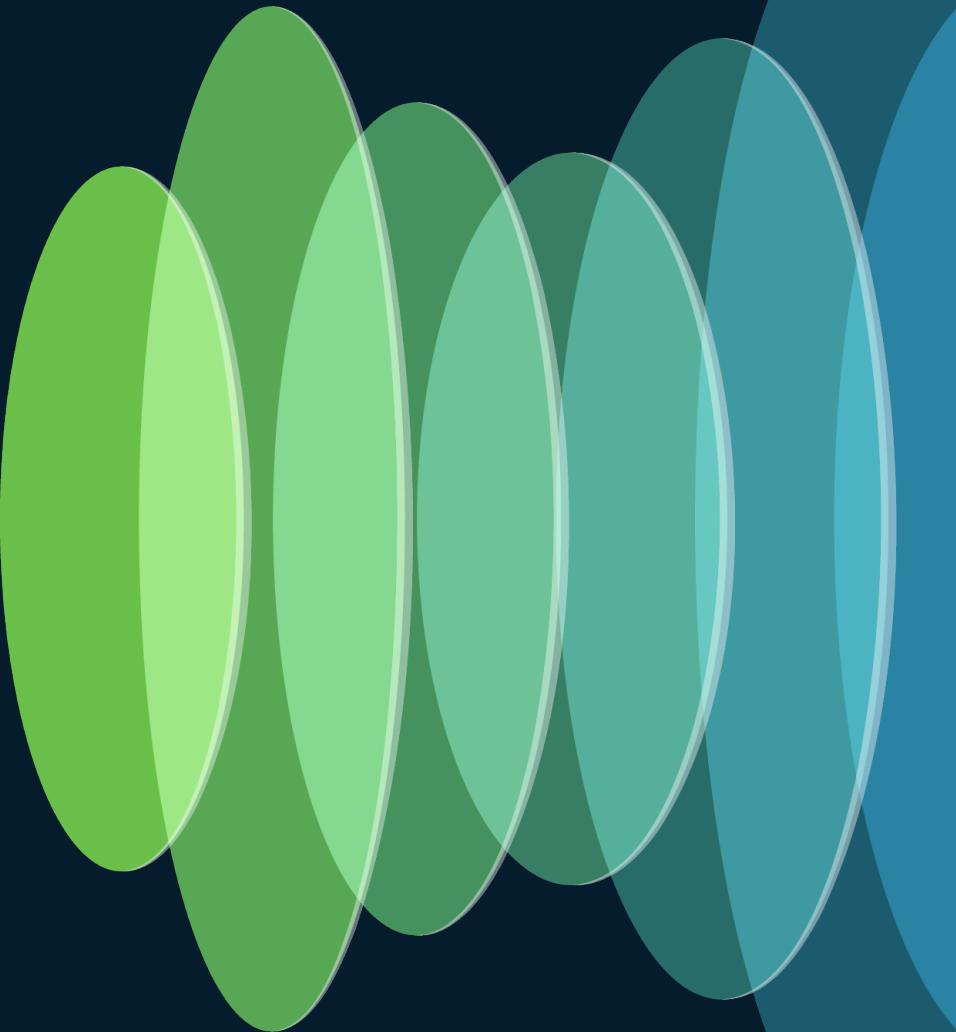
iosxe_cli (Resource)

This resources is used to configure arbitrary CLI commands. This should be considered a last resort in case YANG models are not available, as it cannot read the state and therefore cannot reconcile changes.

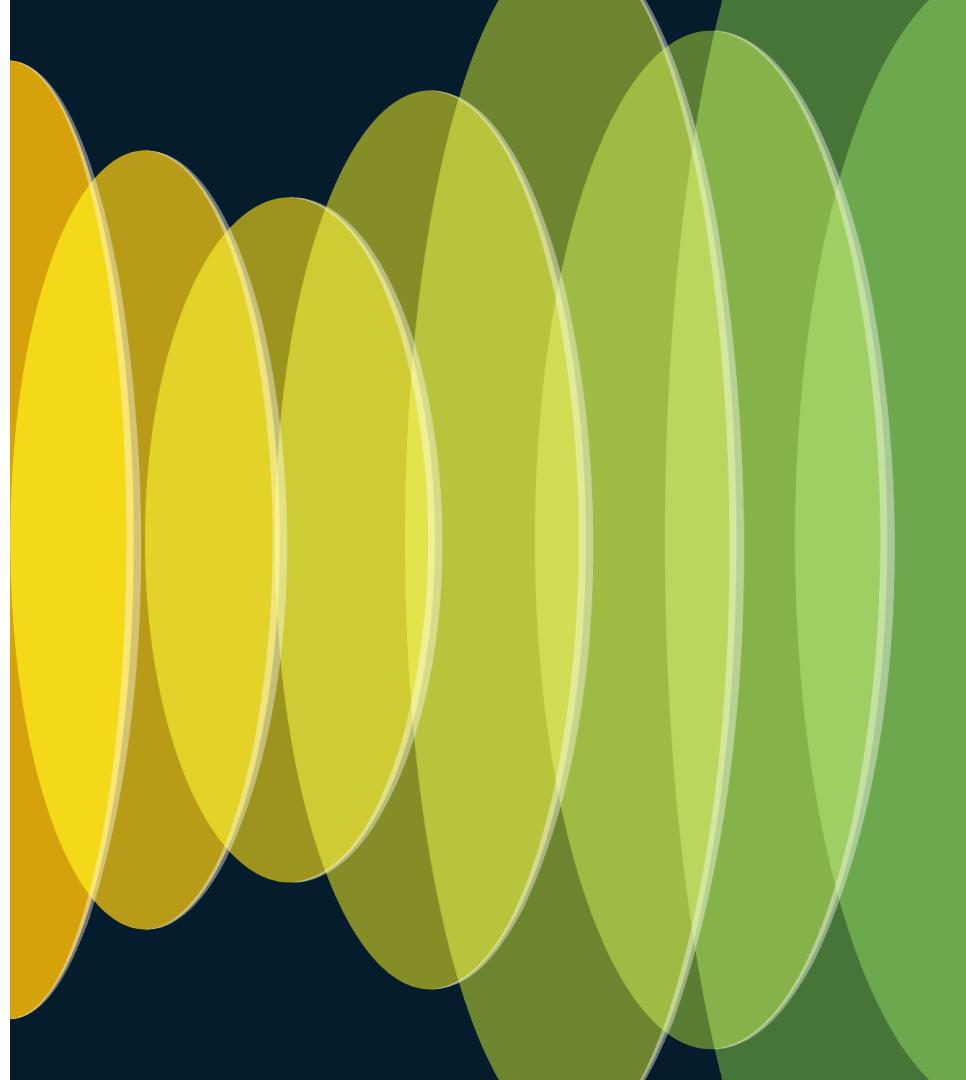
Example Usage

```
resource "iosxe_cli" "example" {
  cli = <<-EOT
  interface Loopback123
  description configured-via-restconf-cli
  EOT
}
```

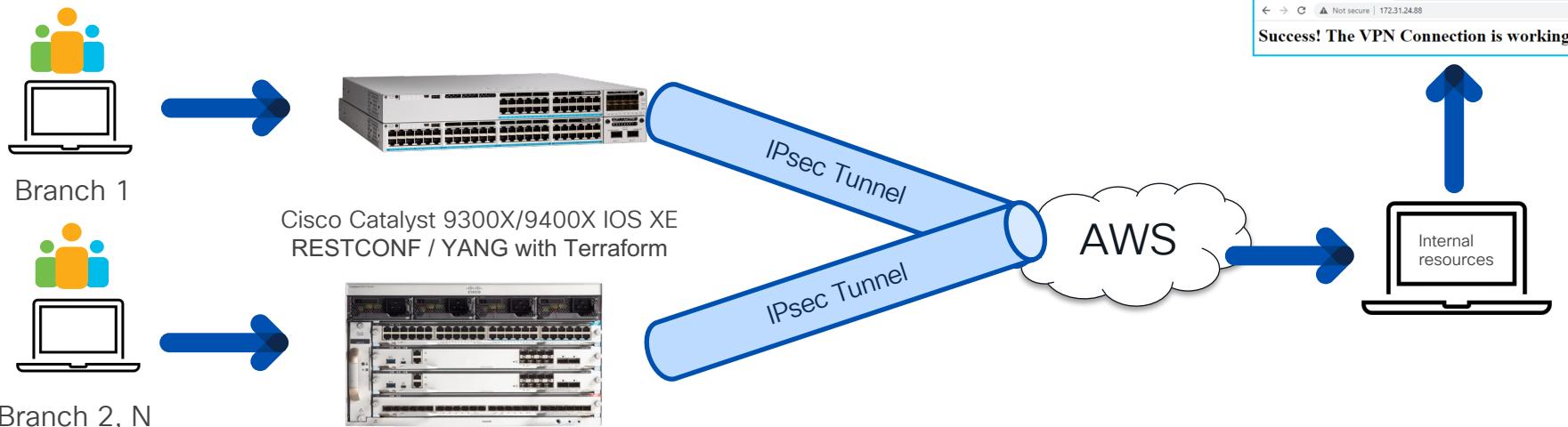
Use Cases & Solutions



IPsec Tunnel + cloud configuration



IPsec Tunnel for employees



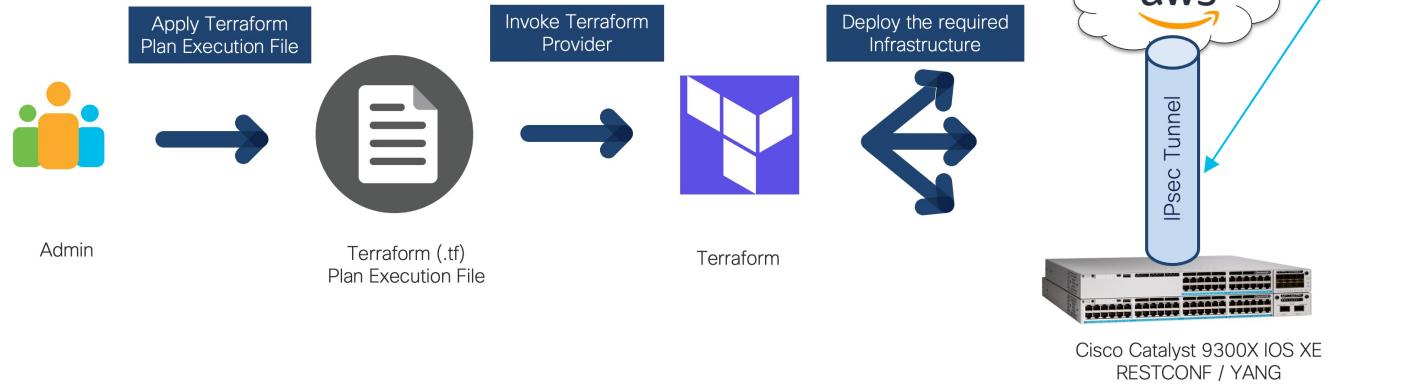
IPsec tunnel + cloud automation with Terraform

Day 0

1. Terraform configures the IPsec tunnel between the 9300X and the cloud service where the internal resources are available
2. Terraform also manages the cloud-native resources including certificate key management and IP subnetting
3. Connections between VPC, VPN, CGW and device certificates, tunnels, and interfaces are created

1. Virtual Private Cloud (VPC)
2. Virtual Private Network (VPN)
3. Customer Gateway (CGW)

1. Tunnel
2. Proposal
3. Policy
4. Keyring
5. Profile
6. IPSEC Transform
7. IPSEC Profile
8. Tunnel Interface



Terraform + Crypto IPsec Demo

The screenshot shows a code editor interface with several tabs open. The main tab contains a Terraform configuration file (`terraform.tf`) with the following content:

```
terraform > terraform.tf > ...
1  # See preview using: terraform plan --var-file="9300X.tfvars"
2  # Run using: terraform apply --var-file="9300X.tfvars" --auto-approve
3
4  terraform {
5      required_providers {
6          iosxe = {
7              source  = "CiscoDevNet/iosxe"
8              version = "0.1.1"
9          }
10     }
11 }
12
13 provider "iosxe" { # variables initialized in variables.tf and values stored in 9300X.tfvars
14     host        = var.host_url
15     insecure    = var.insecure
16     device_username = var.device_username
17     device_password = var.device_password
18 }
19
20 # crypto all
21 resource "iosxe_rest" "crypto_example_post" {
22     method = "PATCH"
23     path   = "/data/Cisco-IOS-XE-native:native/crypto"
24     payload = jsonencode(
25
26     {
27         "Cisco-IOS-XE-native:crypto": {
28             "Cisco-IOS-XE-crypto:ikev2": {

```

Below the code editor, there are tabs for PROBLEMS, OUTPUT, TERMINAL, and DEBUG CONSOLE. To the right, a terminal window is open with the command `zsh`. On the far right, there are several floating panes displaying a detailed tree structure of resources and their properties.

How can I access this code?

 **terraform-ipsec-tunnel-to-cloud** Public

 Pin  Unwatch 1

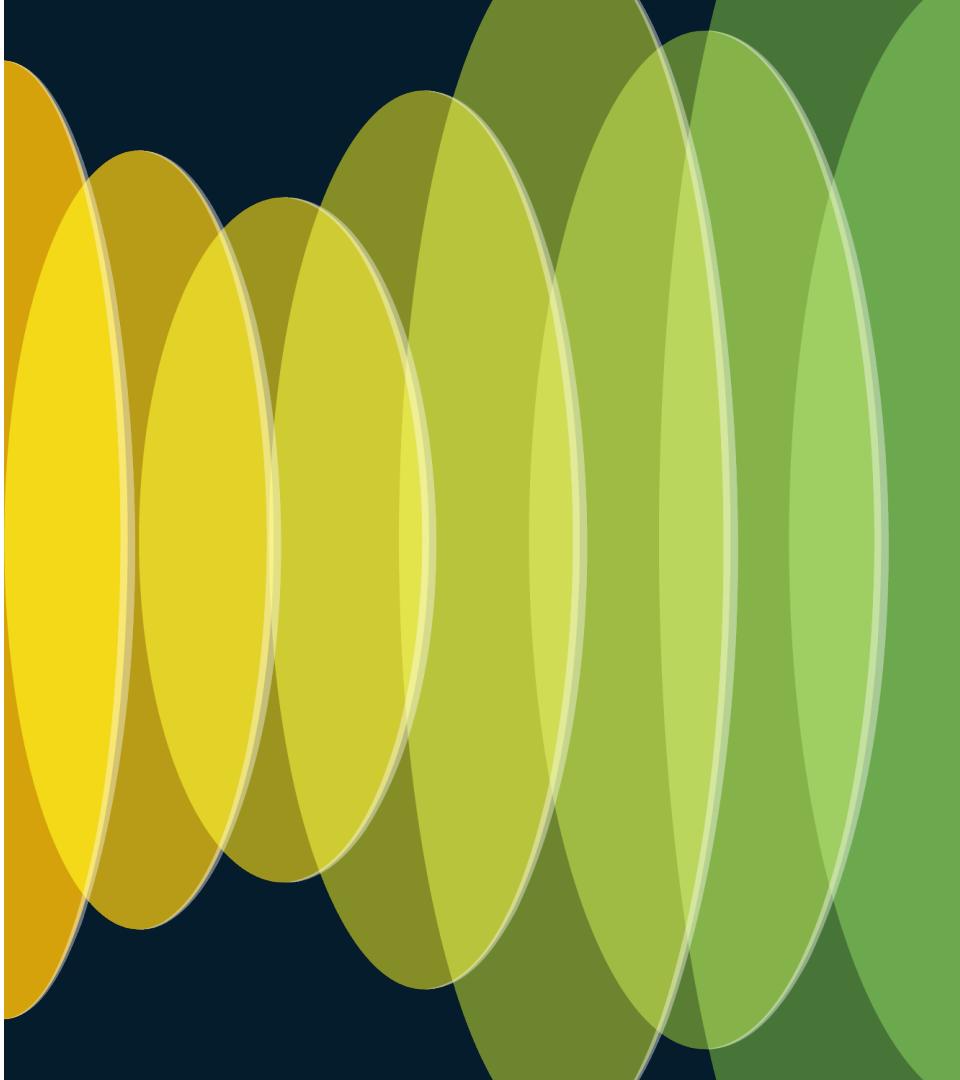
 main  1 Branch  0 Tags  Go to file  Add file  Code

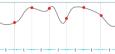
 sdeweese	Create variables.tf	9ee0a62 · 10 months ago	 4 Commits
	9300X.tf	Create 9300X.tf	10 months ago
	aws.tf	Create aws.tf	10 months ago
	outputs.tf	Create outputs.tf	10 months ago
	variables.tf	Create variables.tf	10 months ago

<https://github.com/sdeweese/terraform-ipsec-tunnel-to-cloud>



Model Driven Telemetry



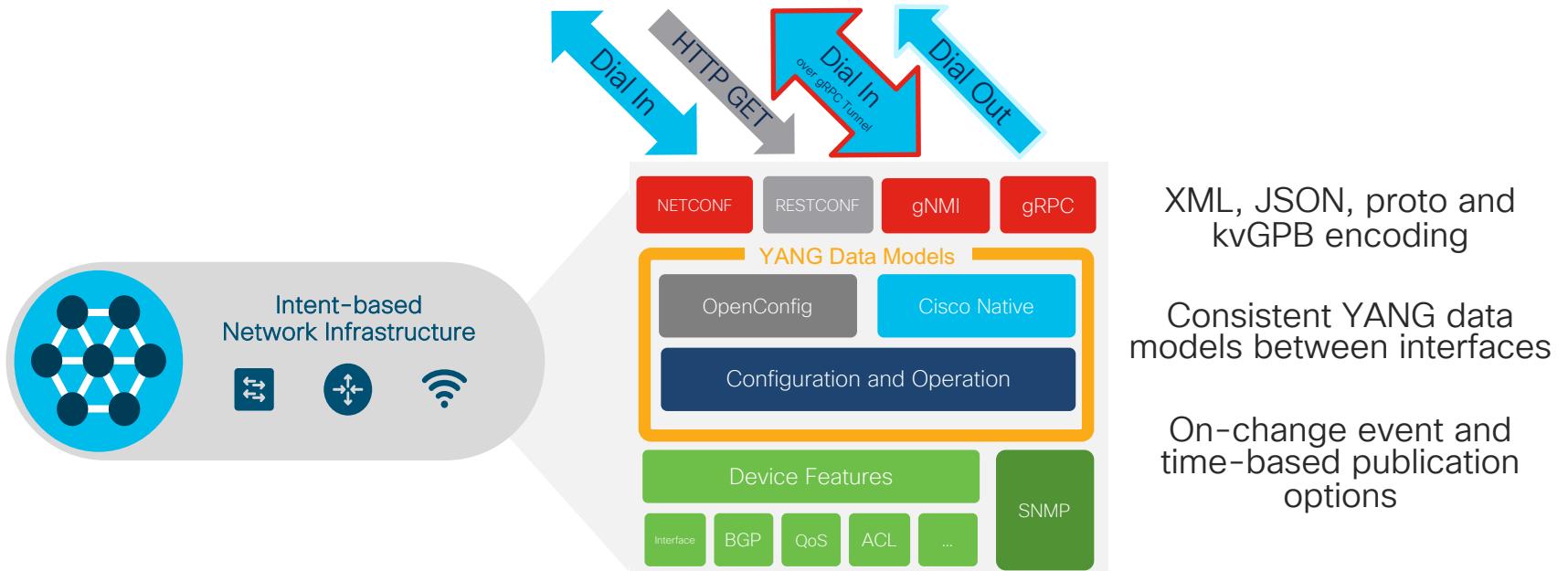


Model Driven Telemetry Interfaces

↔ Dial In: Collector establishes a connection to the device then subscribes to telemetry (pub/sub)

← Dial Out: Telemetry is pushed from the device to the collector based off configuration (push)

Publication / Subscription



Updated TIG_MDT container now available!

Recently updated!

```
docker pull jeremycohoe/tig_mdt
docker run -ti -p 3000:3000 -p 57500:57500 jeremycohoe/tig_mdt
```

Cisco IOS XE Devices



Collector/Receiver
Decodes to text

Storage
Time Series Database

Monitoring
and Visualizations



Upgrade coming to Telegraf, Influx, and Grafana Model Driven Telemetry (TIG_MDT) Docker container

Making it easier to consume telemetry in production

Upgraded Telegraf, InfluxDB, and Grafana tools
Additional dashboards for

Device Health, Wireless Client, Wireless AP, RF etc
Examples for device CLI configuration for telemetry
Details of scale and data storage requirements



https://hub.docker.com/r/jeremycohoe/tig_mdt
https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/179/b_179_programmability_cg/m_179_prog_ietf_telemetry.html

<https://github.com/jeremycohoe/cisco-ios-xe-mdt>



Create MDT Subscriptions using Terraform



Use Terraform to manage the gRPC Dial-Out Telemetry subscriptions

https://registry.terraform.io/providers/CiscoDevNet/iosxe/latest/docs/resources/mdt_subscription

Example files: <https://github.com/jeremycohoe/cisco-ios-xe-mdt/tree/master/sustainability>

iosxe_mdt_subscription (Resource)

This resource can manage the MDT Subscription configuration.

Example Usage

```
resource "iosxe_mdt_subscription" "example" {
  subscription_id      = 101
  stream               = "yang-notif-native"
  encoding             = "encode-kvvpb"
  source_vrf           = "Mgmt-vrf"
  source_address        = "1.2.3.4"
  update_policy_on_change = true
  filter_xpath          = "/ios-events-ios-xe-oper:ospf-neighbor-state-ch
  receivers = [
    {
      address  = "5.6.7.8"
      port     = 57600
      protocol = "grpc-tcp"
    }
  ]
}
```

```
variable "xpaths" {
  default = {
    2024001 = {
      xpath = "/environment-sensors"
    },
    2024002 = {
      xpath = "/oc-platform:components"
    },
    2024003 = {
      xpath = "/platform-ios-xe-oper:components/component"
    },
    2024004 = {
      xpath = "/platform-ios-xe-oper:components/component/platform-properties/platform-property"
    },
    2024005 = {
      xpath = "/poe-oper-data/poe-module"
    },
    2024006 = {
      xpath = "/poe-oper-data/poe-port-detail"
    },
    2024007 = {
      xpath = "/poe-oper-data/poe-stack"
    },
    2024008 = {
      xpath = "/poe-oper-data/poe-switch"
    }
  }
}

provider "iosxe" {
  username = var.host_username
  password = var.host_password
  url      = var.host_url
}

variable host_username {
  type = string
  default = "admin"
}

iosxe_mdt_subscription.example["2024007"][: Modifying... [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024007]
iosxe_mdt_subscription.example["2024008"][: Modifying... [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024008]
iosxe_mdt_subscription.example["2024004"][: Modifying... [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024004]
iosxe_mdt_subscription.example["2024005"][: Modifying... [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024005]
iosxe_mdt_subscription.example["2024001"][: Modifications complete after @s [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024001]
iosxe_mdt_subscription.example["2024005"][: Modifications complete after @s [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024005]
iosxe_mdt_subscription.example["2024006"][: Modifications complete after @s [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024006]
iosxe_mdt_subscription.example["2024003"][: Modifications complete after @s [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024003]
iosxe_mdt_subscription.example["2024007"][: Modifications complete after @s [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024007]
iosxe_mdt_subscription.example["2024002"][: Modifications complete after @s [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024002]
iosxe_mdt_subscription.example["2024004"][: Modifications complete after @s [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024004]
iosxe_mdt_subscription.example["2024008"][: Modifications complete after @s [id=Cisco-IOS-XE-mdt-cfg:mdt-config-data/mdt-subscription=2024008]

Apply complete! Resources: 0 added, 8 changed, 0 destroyed.
auto@pod29-xelab:~/terraform/sustainability$ terraform apply
```

CISCO Live!

Terraform for gRPC Dial-Out telemetry management

1. Modify variables to match your source and destination
 2. Run “terraform init” to ensure your terraform is updated and will use the proper providers
 3. Run “terraform plan” to confirm the config to add is correct
 4. Run “terraform apply” to modify the device config
 5. Validate that the correct config was added to your device

Example TF file

```
1  terraform {  
2    required_providers {  
3      iosxe = {  
4        source  = "CiscoDevNet/iosxe"  
5        version = ">= 0.5.1"  
6      }  
7    }  
8  }  
9  
10 provider "iosxe" {  
11   username = "developer"  
12   password = "Cisco12345"  
13   url      = "https://198.18.1.21"  
14 }  
15  
16 variable source_address {  
17   type = string  
18   default = "1.1.1.1"  
19   description = "Source address"  
20 }  
21  
22 variable receiver_ip {  
23   type = string  
24   default = "1.1.1.1"  
25   description = "Receiver IP"  
26 }  
27  
28 variable receiver_port {  
29   type = string  
30   default = "57500"  
31   description = "Port to send data to"  
32 }
```



```
1  resource "iosxe_mdt_subscription" "cpu_subs" {  
2    for_each          = var.cpu_subscriptions  
3    subscription_id  = each.key  
4    stream            = "yang-push"  
5    encoding           = "encode-kvvpb"  
6    source_vrf         = "Mgmt-vrf"  
7    update_policy_periodic = var.cpu_periodic  
8    source_address     = var.source_address  
9    filter_xpath       = each.value.xpath  
10   receivers = [  
11     {  
12       address  = var.receiver_ip  
13       port     = var.receiver_port  
14       protocol = "grpc-tcp"  
15     }  
16   ]  
17 }  
18  
19 variable cpu_periodic {  
20   type = string  
21   default = "100"  
22   description = "Short update interval"  
23 }  
24  
#CPU.tf  
26 variable "cpu_subscriptions" {  
27   default = {  
28     100 = {  
29       xpath = "/process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-seconds"  
30     },  
31     101 = {  
32       xpath = "/process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/one-minute"  
33     },  
34     102 = {  
35       xpath = "/process-cpu-ios-xe-oper:cpu-usage/cpu-utilization/five-minutes"  
36     }  
37   }  
38 }
```



Create three MDT subscriptions for various CPU data

<https://github.com/jeremycohoe/cisco-ios-xe-dcloud-terraform/tree/main>

Resources

Catalyst Center & Cisco IOS XE Programmability Sessions at Cisco Live Las Vegas 2024

Sunday June 2

Monday June 3

Tuesday June 4

Wednesday June 5

Thursday June 6

9:00-1:00PM **LTCR-CRT-1100:** A Hands-On Preparation for the DevNet Associate Exam (Palmer/Quinn/Kareem)

8:00-9:00AM **BRKOPS-2548:** Network Troubleshooting Using Cisco Catalyst Center APIs (Gabi)

9:00-9:45AM **SKILLS-1770:** Introduction to Cisco Catalyst SDWAN (Anna & Alex)

10:00-10:45AM **SKILLS-1771:** Automation with Cisco Catalyst SDWAN (Tomasz & Alex)

1:00-1:45PM **DEVNET-1283:** Programmability, Automation Model Driven Telemetry on Cisco IOS XE with a dash of YANG Suite (Story)

1:00-1:45PM **IBOOPS-2882:** Let's Talk about Catalyst Center Integrations (Gabi)

4:00-4:45PM **DEVWKS-2031:** Test Automation with Cisco Catalyst 9000 Virtual Switch (Jeremy)

10:00-10:45AM **SKILLS-1110:** Configure IOS XE using CLI (Jeremy & Story)

11:00-11:45AM **SKILLS-1111:** Configure IOS XE using Automation (Jeremy & Story)

12:00-12:45PM **DEVNET-1087:** Cisco Catalyst Center Platform: APIs, Event Notifications, Integrations, and DevOps Resources (Gabi)

12:00-12:45PM **SKILLS-1770:** Introduction to Cisco Catalyst SDWAN (Anna & Alex)

1:00-1:45PM **SKILLS-1771:** Automation with Cisco Catalyst SDWAN (Tomasz & Alex)

2:30-3:30PM **BRKDEV-2017:** gRPC, gNMI, gNOI... Oh My! An Enterprise Network Automation Journey (Jeremy+Story)

4:00-4:45PM **DEVWKS-2031:** Test Automation with Cisco

10:00-10:45AM **SKILLS-1660:** Introduction to Catalyst Center (Gabi & Adrian)

11:00-11:45AM **SKILLS-1661:** Introduction to Catalyst Center Platform (Gabi & Adrian)

10:30-11:30AM **BRKOPS-2032:** Cisco Catalyst Center and ITSM Workflows: CMDB, Incident Management, and SWIM (Gabi)

1:00-1:45PM **DEVWKS-2042:** Becoming a Cisco Catalyst IOS XE Terraform Expert (Story)

3:00-3:45PM **DEVNET-3000:** ChatBot for Catalyst Center – on Open-Source AI based Bot (Gabi)

3:00-3:45PM **SKILLS-1110:** Configure IOS XE using CLI (Jeremy & Story)

4:00-4:45PM **SKILLS-1111:** Configure IOS XE using Automation (Jeremy & Story)

The Celebration !

9:00-9:45AM **SKILLS-1660:** Introduction to Catalyst Center (Gabi & Adrian)

10:00-10:45AM **SKILLS-1661:** Introduction to Catalyst Center Platform (Gabi & Adrian)

Session Levels:
Beginner
Intermediate
Advanced

All week in World of Solutions LABPRG-2004:
Efficiently monitoring device statistics in real-time using gRPC Dial-out with IOS XE

API White Paper

Programmability and auto... ▾ Q

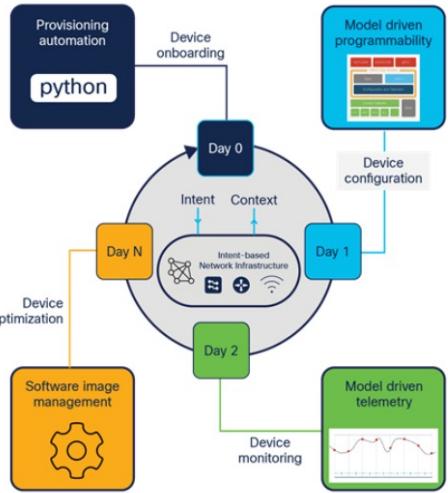
Table of Contents

- Programmability and automation...
- Day 0: Provisioning automation
- Day 1: Model-driven programmability
- Day 2: Model-driven telemetry
- Day N: Device optimization
- Cisco IOS XE operational consistency
- Yet Another Next Generation (Y...+
 - Day 1: Model-driven program...
 - Tooling: Cisco YANG Suite
 - Day 2: Model-driven telemetry
 - Day N: Device optimization
 - Conclusion
 - Additional resources
 - Blogs

Products & Services / Switches / Campus LAN Switches - Access / Cisco Catalyst 9300 Series Switches /

Catalyst Programmability and Automation

Pre-boot Execution
Environment (iPXe)
Zero touch provisioning
VM automation



Website: <https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9300-series-switches/nb-06-catalyst-programmability-automation-wp.html>

PDF: <https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9300-series-switches/nb-06-catalyst-programmability-automation-wp.pdf>



MDT White Paper

The Model Driven Telemetry White Paper includes examples, use cases and tooling related to telemetry

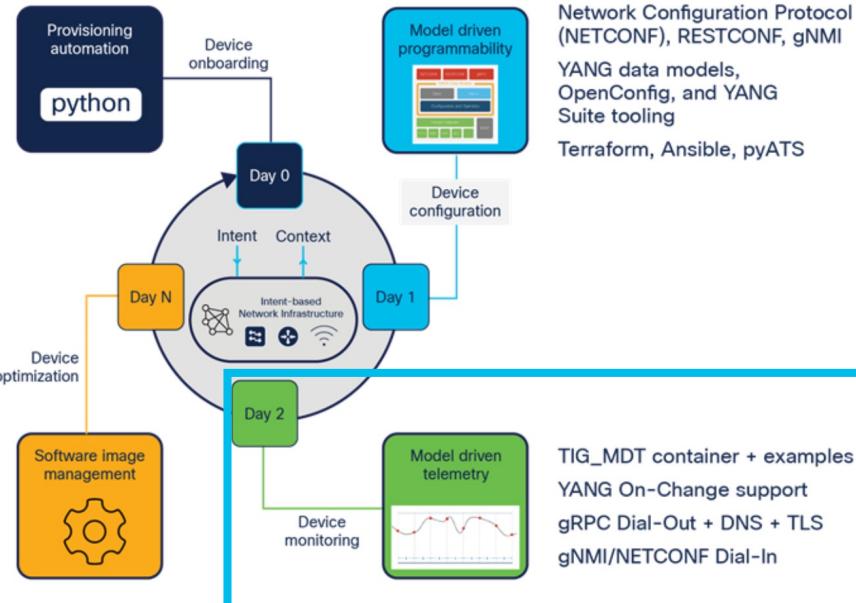
Introduction to Cisco IOS ... ▾ Q

Table of Contents

- Introduction to Cisco IOS XE
- Introduction to telemetry
- Benefits of model driven tele...
- Network monitoring challeng...
- Architecture and databases
- Dial-in and dial-out MDT +
- Publication notification opti...
- YANG data modeling language +
- Benchmarking and comparis... +
- Cisco controller solutions +
- Cloud solutions +
- Tooling +
- Dashboarding and validation +
- Configuration examples +
- Telemetry configuration man... +
- Troubleshooting and validati... +
- Best practices and lessons l... +
- Conclusion
- Resources

Pre-boot Execution
Environment (iPXE)
Zero touch provisioning
VM automation

gNOI cert/os/reset proto
Guest shell + NETCONF
CentOS 8 Python 3
Application hosting with Docker
CLI to YANG



Website: <https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9300-series-switches/model-driven-telemetry-wp.html>
PDF: <https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9300-series-switches/model-driven-telemetry-wp.pdf>



<http://cs.co/mdtwp>



<http://cs.co/mtpwppdf>



Programmability Configuration Guide

Preface

New and Changed Information

✓ Provisioning

Zero-Touch Provisioning

iPXE

✓ Shells and Scripting

Guest Shell

Python API

EEM Python Module

✓ Model-Driven Programmability

NETCONF Protocol

RESTCONF Protocol

NETCONF and RESTCONF Service-Level ACLs

gNMI Protocol

gRPC Network Operations Interface

gNMI Dial-Out Using the gRPC Tunnel Service

Model Based AAA

Model-Driven Telemetry

In-Service Model Update

✓ Application Hosting

Application Hosting

ThousandEyes Enterprise Agent



https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/prog/configuration/1714/b_1714_programmability_cg.html

Programmability Website

The one-stop-shop for Cisco IOS XE Programmability resources including videos, white papers, labs and more!

The screenshot shows the Cisco DevNet website with a blue header bar. The main content area has a dark blue background with white text. It features a title 'Cisco IOS XE Programmability and Automation' and a sub-section 'Programmability options with IOS XE'. Below this, there are two sections: 'Provisioning' and 'Configuration management'. Each section includes a small icon and a brief description. At the bottom of the page, there are three cards: 'LIVE STREAM Terraform + IOS XE | A Match Made on the Internet', 'WEBINAR Automating Catalyst IOS XE', and 'WHITE PAPER Catalyst Programmability and Automation'.

- Community Forum
- IOS XE FAQ
- White Papers
- Code Exchange
- IOS XE Docs & Guide
- Learning Tracks and Labs
- Sandboxes
 - ... and more !



<https://developer.cisco.com/iosxe/>

dCloud Programmability

<https://dcloud.cisco.com>

"Cisco Catalyst 9000 IOS XE Programmability & Automation Lab v1"

<https://dcloud2.cisco.com/demo/catalyst-9000-ios-xe-programmability-automation-lab-v1>

Use Cases:

EVPN:

- Ansible with CLI deployment of EVPN solutions
- EVPN management over RESTCONF/YANG with Postman
- Declarative EVPN fabric management with Terraform

Model Driven Telemetry

Telemetry configuration with CLI and YANG Suite Collection with TIG_MDT container and tooling

YANG Programmability

YANG Suite tooling and integrations to YANG API's Ansible integrations

Tooling and Integrations

YANG Suite

- NETCONF/RESTCONF/gNMI API
 - Ansible integration
- NETCONF/gNMI Dial-In Telemetry
- gRPC Dial-Out Telemetry receiver

Telemetry

- TIG stack in Docker
- Grafana dashboard for device health

Postman / RESTCONF

- EVPN fabric API calls

Terraform/RESTCONF

- Declarative EVPN fabric management

Ansible

- EVPN solution enablement using CLI

Ubuntu VM Details:

Syslog receiver from all switches
TFTP config backup
See slide

Windows VM Details

VS Code

Terraform @ folder

Ansible @ folder

Chrome browser

YANG Suite, Grafana

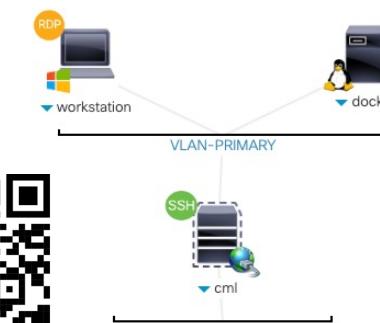
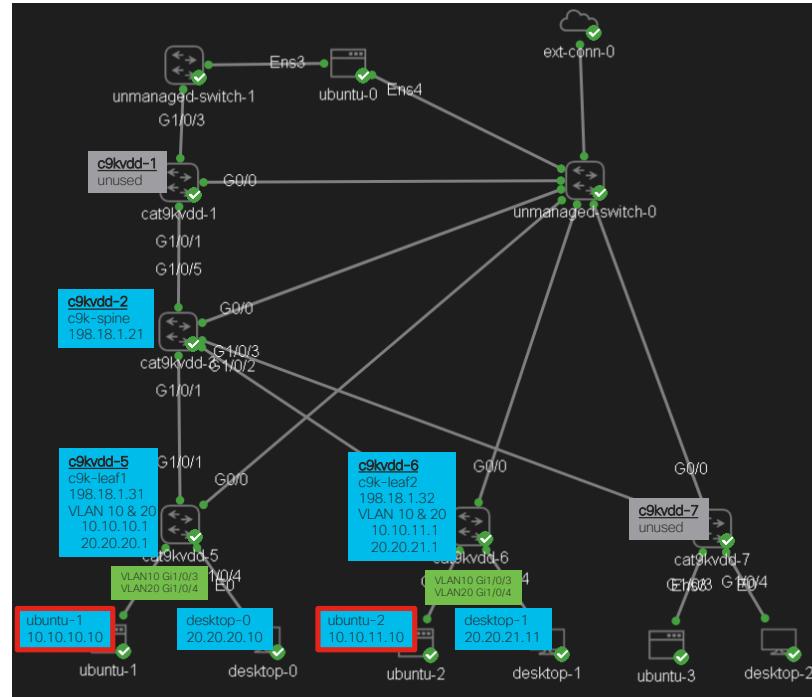
Bash/PS/Cmd shells

SSH into C9K or Ubuntu

Postman

Workspace for EVPN

3x C9K Virtual Switch



VLAN1	c9k-spine	IP: 198.18.1.21 developer / C1sco12345
c9k-leaf1		IP: 198.18.1.31 developer / C1sco12345
c9k-leaf2		IP: 198.18.1.32 developer / C1sco12345
c9kvdd-1 - unconfigured		
c9kvdd-7 - unconfigured		

Cisco IOS XE - Reservable Virtual Sandbox

This DevNet reservable sandbox has IOS XRv + N9Kv + IOS XE
The Catalyst 8000V within the DevBox is enabled for Day 0/ZTP usecases

<https://devnetsandbox.cisco.com>

“IOS XE on Cat8kv”

<https://devnetsandbox.cisco.com/DevNet/catalog/IOS%20XE%20on%20Cat8kv>

The “Enterprise Networking” Learning Labs contains guides for the supported usecases

<https://developer.cisco.com/learning/>



IOS XE on Cat8kv



The IOS XE on Catalyst 17.9.2 Sandbox offers developers access to an IOS XE device running the latest IOS XE code release available on cisco.com. Here you can test out the newest programmability...

Networking reservable



Launch

Explore YANG data models and NETCONF to query and configure IOS XE network devices. Then use Ansible to manage configurations on your IOS XE devices, covering the fundamentals of Ansible. Lastly, enable...

① 6 hrs - 4 modules - 12 labs

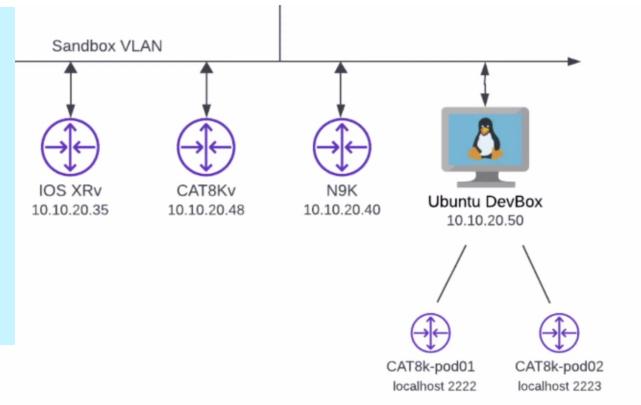
CISCO Live!

#CiscoLive

DEVWKS-2042

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

96



Sandbox Capabilities:

TIG_MDT Telemetry
YANG Suite API
Terraform + Ansible
ZTP & Guest Shell
... and more

Table of Contents

- Overview
- Network Topology
- Grafana Dashboard
- Access Details
- Additional Resources
- VPN Access

Catalyst 8000v Credentials

host	port	username	password
10.10.20.48	22	developer	Cisco12345
10.10.20.48	830	developer	Cisco12345
10.10.20.48	443	developer	Cisco12345

Developer Box Credentials

host	port	username	password
10.10.20.50	22	developer	Cisco12345
telnet localhost (CAT8k-1)	2222	developer	Cisco12345

Cisco IOS XE - Always On Virtual Sandbox

The “IOS XE on Cat8kv Always On” virtual sandbox requires no reservation

The c8kv can be accessed with SSH, NETCONF, and RESTCONF

Hostname: `sandbox-iosxe-latest-1.cisco.com`

Username: `admin` Password: `C1sco12345`

SSH port 22, NETCONF port 830, RESTCONF HTTPS

<https://devnetsandbox.cisco.com>

`devnetsandboxiosxe.cisco.com`

<https://devnetsandbox.cisco.com/DevNet/catalog/ios-xe-always-on>

The “Enterprise Networking” Learning Labs contains guides for the supported usecases

<https://developer.cisco.com/learning/>

IOS XE on Cat8kv AlwaysOn



This AlwaysOn sandbox offers developers access to an IOS XE device running the latest IOS XE code release available on cisco.com (currently 17.11.x). Here you can test out the newest programmability feature...

Always-On



Launch

CISCO Live!

Access Details:

Developers and network engineers access the IOS XE on Catalyst 17.11.x sandbox directly using the following information:

- Cat8000v Host
 - Address: `sandbox-iosxe-latest-1.cisco.com`
 - Username: `admin`
 - Password: `C1sco12345`
 - NETCONF port: 830
 - gRPC telemetry port: 57500
 - ssh port: 22

Learning Track Enterprise Networking

Explore YANG data models and NETCONF to query and configure IOS XE network devices. Then use Ansible to manage configurations on your IOS XE devices, covering the fundamentals of Ansible. Lastly, enable...

6 hrs - 4 modules - 12 labs

Accessing the Sandbox

The screenshot shows the DevNet Environment Overview page. At the top, it displays the environment name 'IOS XE on Cat8kv AlwaysOn', its status as 'Active', and the owner 'Story DeWeese'. It shows an uptime of '0m 32s' and time remaining of '2d 23h'. Below this is a navigation bar with tabs for 'Activity', 'Cloud', 'Quick Access', and 'Workflows'. A sidebar on the left lists '0 Resources' and includes a search bar. The main content area features a 'Table of Contents' with links to 'Overview', 'Access Details', and 'Additional Information'. The 'Overview' section contains text about the Catalyst Always-On sandbox, mentioning shared access to an IOS XE device running version 17.12.02, and highlights programmability features like YANG Data Models and NETCONF. It also lists RESTCONF and NETCONF ports, and gRPC telemetry port.

devnetsandbox.cisco.com

Access Details:

Developers and network engineers access the **IOS XE on Catalyst 17.12.02** sandbox directly using the following information:

- **Cat8000v Host:**
 - **Public URL:** devnetsandboxiosxe.cisco.com
 - **Username:** admin
 - **Password:** C1sco12345
 - **RESTCONF port:** 443
 - **NETCONF port:** 830
 - **gRPC telemetry port:** 57500
 - **ssh port:** 22
 - **Example Connection:** ssh admin@devnetsandboxiosxe.cisco.com

Using the Sandbox

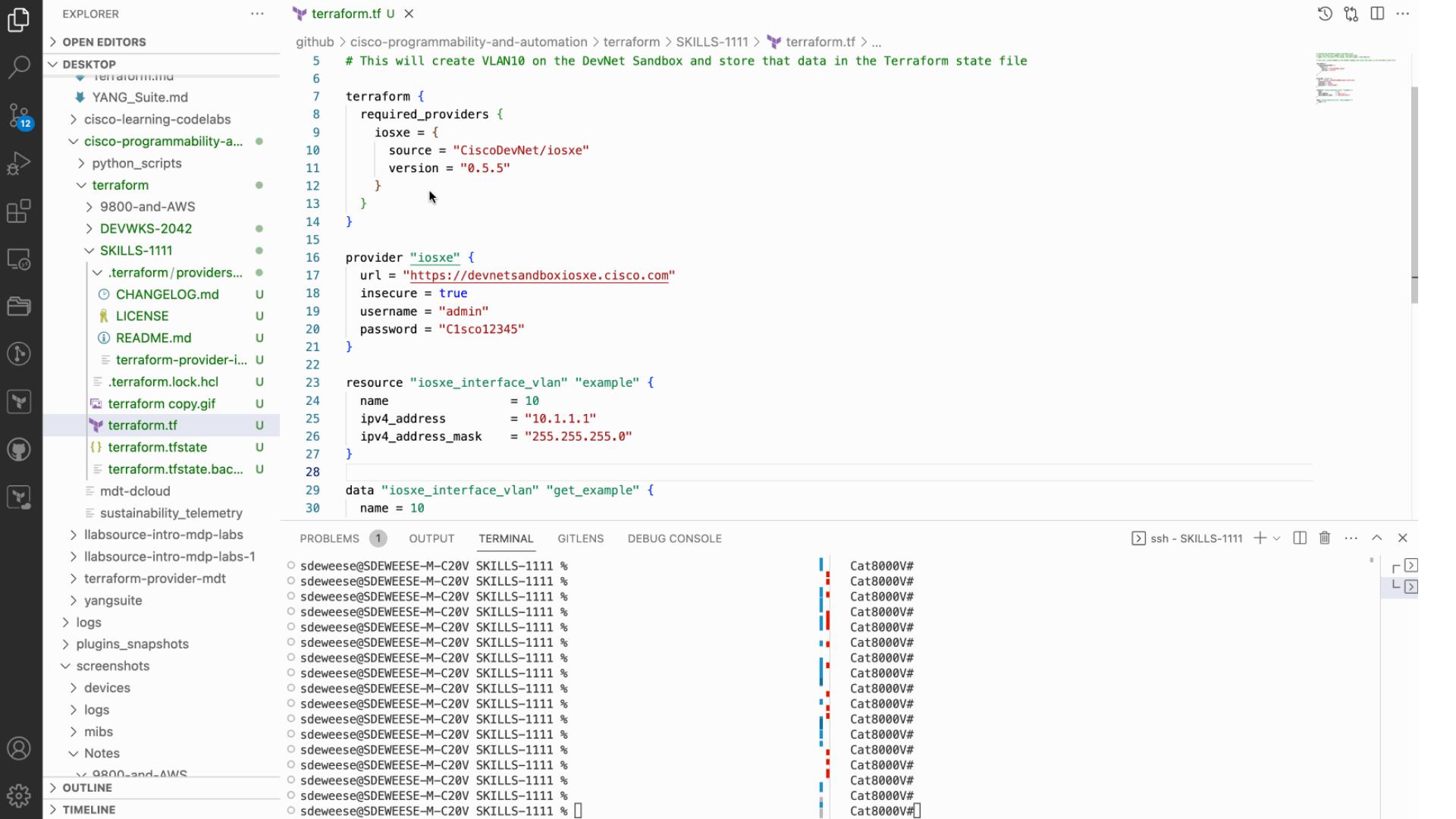
```
1 # initialize terraform using 'terraform init'
2 # plan out what will happen using 'terraform plan'
3 # apply this terraform file using 'terraform apply --auto-approve'
4
5 # This will create VLAN10 on the DevNet Sandbox and store that data in the Terraform state file
6
7 terraform {
8     required_providers {
9         iosxe = {
10            source = "CiscoDevNet/iosxe"
11            version = "0.5.5"
12        }
13    }
14 }
15
16 provider "iosxe" {
17     url = "https://devnetsandboxiosxe.cisco.com"
18     insecure = true
19     username = "admin"
20     password = "Cisco12345"
21 }
22
23 resource "iosxe_interface_vlan" "example" {
24     name          = 10
25     ipv4_address = "10.1.1.1"
26     ipv4_address_mask = "255.255.255.0"
27 }
28
29 data "iosxe_interface_vlan" "get_example" {
30     name = 10
31 }
```

Before Terraform: no VLAN

```
Cat8000V#sh ip int brief
Interface          IP-Address      OK? Method Status      Protocol
GigabitEthernet1   10.10.20.48   YES NVRAM up       up
GigabitEthernet2   10.255.255.1  YES other up       up
GigabitEthernet3   unassigned     YES NVRAM administratively down down
Loopback0          10.0.0.1      YES NVRAM up       up
Loopback10         unassigned    YES unset up       up
Loopback100        172.16.100.1 YES other up       up
VirtualPortGroup0 192.168.1.1   YES NVRAM up       up
```

After Terraform: VLAN 10 Created

```
Cat8000V#sh ip int br
Interface          IP-Address      OK? Method Status      Protocol
GigabitEthernet1   10.10.20.48   YES NVRAM up       up
GigabitEthernet2   10.255.255.1  YES other up       up
GigabitEthernet3   unassigned     YES NVRAM administratively down down
Loopback0          10.0.0.1      YES NVRAM up       up
Loopback1          10.0.1.1      YES manual up      up
Loopback10         unassigned    YES unset up       up
Loopback100        172.16.100.1 YES other up       up
VirtualPortGroup0 192.168.1.1   YES NVRAM up       up
Vlan10             10.1.1.1      YES other down    down
```



Cisco University (Cisco U) part of L&D

u.cisco.com or <https://u.cisco.com>

<https://u.cisco.com/search/tutorial?query=Story%20DeWeese,%20Jeremy%20Coho,%20not%20berry>

Direct link to Tutorial, requires login to u.cisco.com first:

1. <https://ondemandlearning.cisco.com/apollo-alpha/tc-iosxe-ztp/pages/1>
2. <https://ondemandlearning.cisco.com/apollo-alpha/tc-terraform-ios-xe/pages/1>
3. <https://ondemandlearning.cisco.com/apollo-alpha/tc-yangsuite-netconf/pages/1>
4. <https://ondemandlearning.cisco.com/apollo-alpha/tc-yangsuite-restconf/pages/1>

Cisco U.

Welcome to Cisco U.

You're in the right place, whether you're looking to earn a certification or gain new skills. In Cisco U, you'll find courses, community, and learning content to help you reach your goals.

[Learn more about Cisco U.](#), or come on in and get started!

[Log in with your Cisco Account ID](#)

[Create a free account](#)

The image displays four cards from the Cisco U platform, each representing a different tutorial:

- TUTORIAL** by Jeremy Cohoe: Cisco IOS XE Zero-Touch Provisioning. Beginner level, 45m duration. Created October 23, 2023.
- TUTORIAL** by Story DeWeese: Using YANG Suite with NETCONF. Intermediate level, 30m duration. Created October 24, 2023.
- TUTORIAL** by Story DeWeese: Using YANG Suite with RESTCONF and gRPC. Intermediate level, 10m duration. Created October 24, 2023.
- TUTORIAL** by Story DeWeese: Cisco IOS XE Terraform Provider. Beginner level, 15m duration. Created February 21, 2023.

CISCO Live!

#CiscoLive

DEWWSK-2042

© 2024 Cisco and/or its affiliates. All rights reserved. Cisco Public

101



Continue your education

CISCO Live!

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand



The bridge to possible

Thank you

CISCO Live!

#CiscoLive